Анализ поточных шифров с помощью решения системы алгебраических уравнений

03, март 2013

DOI: 10.7463/0313.0546388

Чиликов А. А., Хузина Э. И.

УДК 003.26

Россия, МГТУ им. Н.Э. Баумана

chilikov@passware.com

x47y47z47@mail.ru

Построение криптографических поточных шифров основано на использовании генератора ключевого потока. Такой генератор с помощью ключа формирует ключевой поток, затем ключевой поток комбинируется определенным образом с исходным сообщением с целью получения зашифрованного сообщения. Отправитель передает зашифрованное сообщение законному получателю, который, имея ключ, формирует ключевой поток, и расшифровывает сообщение.

Согласно результатам исследований [6] при значениях параметров ключевого потока больше чем 4, атаки поточных шифров становятся неэффективными. В данной работе проведен анализ ключевого потока поточного шифра с помощью решения системы алгебраических уравнений при известных некоторых данных и найдены максимальные параметры генератора ключевого потока, при которых поточный шифр поддается анализу программными средствами.

Криптосистема или шифр – это кортеж из пяти элементов (P, C, K, E, D) , таких что [3]:

- P – конечное множество исходных текстов;
- C – конечное множество шифротекстов;
- K – конечное множество символов ключевого потока;
- $E : K \times P \rightarrow C$ алгоритм кодирования;

- $D : K \times C \rightarrow P$ алгоритм декодирования;

Для любого $K \in K$, выполняется $D(K, E(K, P)) = P$ для всех $P \in P$.

Поточные шифры преобразуют исходный текст в шифротекст по одному биту за операцию. Разновидность поточного шифра – схема одноразового блокнота. Для схемы одноразового блокнота $P = C = K = \{0,1\}^n$, при $n \geq 1$, алгоритм кодирования $E_K(P) = K \oplus P$, где $\oplus$ – операция покомпонентного XOR, алгоритм декодирования $D_K(C) = K \oplus C$. Доказано, что схема одноразового блокнота обладает абсолютной криптографической стойкостью, если K используется один раз и выбран из $\{0,1\}^n$ [2]. Таким образом, если злоумышленник знает только C , он не сможет получить какую-либо информацию о P или K , даже при наличии неограниченных времени, памяти и вычислительных ресурсов.

Работа посвящена анализу поточного шифра, созданного с помощью схемы одноразового блокнота. Генератор ключевого потока инициализируют ключом меньшего размера, чем сам ключевой поток. Генератор ключевого потока вырабатывает псевдослучайный поток битов длины, равной длине исходного текста; создается псевдослучайная ключевая последовательность. Для анализа шифротекста в этом случае необходимо знать ключ и сгенерировать ключевой поток, использованный при шифровании. Выполнив покомпонентный XOR ключевого потока и шифротекста, можно получить исходное сообщение.

Генератор ключевого потока состоит из следующих компонентов [2]:

- внутренне состояние;
- K – конечное множество возможных внутренних состояний;
- функция обновления $Y : K \rightarrow K$;
- конечное множество Z , называемое алфавитом ключевого потока;
- функция выхода $f : K \rightarrow Z$.

Внутреннее состояние $S_0 \in K$ инициализируют ключом (или начальным состоянием), затем выполняют следующие операции:

- выходной бит $Z_t \in Z$ вычисляется как $Z_t = f(S_t)$;

- внутреннее состояние обновляется $S_{t+1} = Y(S_t)$.

Такт – это период времени, в течение которого выполняются две вышеперечисленные операции. В такте t производится выходной бит Z_t и S_t меняется на S_{t+1} .

Поточный шифр, основанный на генераторе ключевого потока - это совокупность кортежа (P, C, K, Z, E, D) и генератора ключевого потока, где [6]:

P – конечное множество, называемое алфавитом исходного текста;

C – конечное множество, называемое алфавитом шифротекста;

K – конечное множество возможных ключей;

Z – конечное множество, называемое алфавитом ключевого потока;

$E : Z \times P \rightarrow C$ алгоритм кодирования;

$D : Z \times C \rightarrow P$ алгоритм декодирования.

Генератор ключевого потока использует ключ $K \in K$ для генерации последовательности $Z_0, Z_1, Z_2, \dots$, называемой ключевым потоком произвольной длины, где $Z_t \in Z$ для всех $t \geq 0$. Для каждого $Z \in Z$, выполняется $D(Z, E(Z, P)) = P$ для всех $P \in P$.

Получатель и отправитель взаимодействуют следующим образом:

- получатель и отправитель обмениваются секретным ключом $K \in K$ по секретному каналу, секретный канал используют для передачи всех необходимых данных;
- пусть $P = (p_0, p_1, p_2, \dots)$ – исходный текст, где $p_t \in P$. Отправитель использует K и генератор ключевого потока для создания ключевого потока $Z_0, Z_1, Z_2, \dots$ и кодирует P в шифротекст $C = (c_0, c_1, c_2, \dots)$, где $c_t = E_{Z_t}(p_t)$, затем посылает C получателю через общедоступный канал;
- получатель, зная секретный ключ, формирует у себя ключевой поток $Z_0, Z_1, Z_2, \dots$;
- получатель, используя ключевой поток $Z_0, Z_1, Z_2, \dots$ декодирует $p_t = D_{Z_t}(c_t)$.

Пусть F – конечное поле; регистр сдвига с линейной обратной связью (РСЛОС) длины n содержит внутреннее состояние размерности n и функцию обратной связи

$$\Lambda(x_1, \dots, x_n) = \sum_{i=1}^n \lambda_{i-1} \cdot x_i \in F[x_1, \dots, x_n].$$

В каждом такте определенный элемент внутреннего состояния становится выходным битом и внутреннее состояние изменяется на новое с помощью Λ . Пусть $S_0 = (z_0, \dots, z_n) \in F^n$ является начальным состоянием и $L(x_1, \dots, x_n) = (x_2, \dots, x_n, \Lambda(x_1, \dots, x_n))$ [2]. Схематичный вид РСЛОС представлен на рисунке 1. В каждый такт выполняются две операции[3]:

- на выход подается определенный элемент;
- изменение S_t на S_{t+1} с помощью функции L :

$$S_{t+1} = L(S_t) = (s_{t+1}, \dots, s_{n+t-1}, \Lambda(S_t)) = (s_{t+1}, \dots, s_{n+t-1}, \sum_{i=0}^{n-1} \lambda_i \cdot s_{t+i}).$$

Функцию L можно идентифицировать с матрицей:

$$LL := \begin{pmatrix} 0 & \dots & \dots & 0 & \lambda_0 \\ 1 & \ddots & & \vdots & \lambda_1 \\ 0 & \ddots & \ddots & \vdots & \vdots \\ \vdots & \ddots & \ddots & 0 & \vdots \\ 0 & \dots & 0 & 1 & \lambda_{n-1} \end{pmatrix}$$

При этом $S_{t+1} = S_t \cdot LL$, также $S_t = S_0 \cdot LL^t$. Матрица LL называется матрицей обратной связи [2].

Рисунок 1 – РСЛОС.

В качестве генератора ключевого потока при создании поточного шифра использован (l, m) -комбайнер. Пусть F – конечное поле, (l, m) -комбайнер состоит из следующих шести компонентов [2]:

- s РСЛОС с длинами $n_1, \dots, n_s$ и матрицами обратной связи $L_1, \dots, L_s$;

- внутреннее состояние $S \in F^m \times F^n$, где $n = n_1 + \dots + n_s$;
- матрица L , размером $n \times n$, над полем F , определена как

$$L := \begin{pmatrix} L_1 & & 0 \\ & \ddots & \\ 0 & & L_s \end{pmatrix}$$

- матрица проекций P , размером $n \times 1$, над полем F ;
- функция изменения состояния памяти $\Psi : F^m \times F^1 \rightarrow F^m$;
- функция выхода $f : F^m \times F^1 \rightarrow F$.

Если $m \geq 1$, тогда комбайнер называется комбайнером с памятью, а при $m = 0$ – простым комбайнером.

Внутреннее состояние (l, m) -комбайнера инициализируем значением: $S_0 = (Q_0, K) \in F^m \times F^n$, где $Q_0 \in F^m$ – биты памяти, $K = (K_{0,1}, \dots, K_{0,s}) \in F^n$ – начальные состояния s РСЛОС, где $K_{0,i}$ – начальное состояние i -го РСЛОС. Содержание РСЛОС обновляется с помощью матрицы L [2].

Во многих случаях используют только некоторые значения в состояниях РСЛОС для вычисления следующего элемента ключевого потока и следующего состояния памяти, определяемые матрицей проекции. Только значения из $K \cdot L^t \cdot P$ вовлечены в вычисления на такте t . Обозначим $K_t = K \cdot L^t \cdot P$ и назовем K_t – вход (l, m) -комбайнера. Обновление состояния памяти происходит через $Q_{t+1} = \Psi(Q_t, K_t)$, а элемент ключевого потока вычисляется как $z_t = f(Q_t, K_t)$. Функция обновления $Y(S_t) = Y(Q_t, K \cdot L^t) = (\Psi(Q_t, K \cdot L^t \cdot P), K \cdot L^{t+1})$ [2]. Схематичный вид (l, m) -комбайнера показан на рисунке 2.

Рисунок 2 – (l, m) -комбайнер.

Найдем первоначальное состояние (ключ) (l,m) -комбайнера. Пусть $K_t \in F^l$ - вход функции выхода f в такте t и (z_t) – ключевой поток. Функция $F_Z: F^{r+1} \rightarrow F$ при $Z \in F^r$ называется Z -функцией, когда при $z_t, \dots, z_{t+r-1} = Z$, F_Z равно нулю на соответствующих входах $K_t, \dots, K_{t+r-1}$. То есть

$$(z_t, \dots, z_{t+r-1}) = Z \Rightarrow F_Z(K_t, \dots, K_{t+r-1}) = 0 \quad [3].$$

Если злоумышленник знает значения $z_{t1}, \dots, z_{t1+r-1}, z_{t2}, \dots, z_{t2+r-1}, \dots, z_{tN}, \dots, z_{tN+r-1}$, то используя перечисленные данные, он может составить следующую систему уравнений из Z -функций [2]:

$$\begin{cases} F_{z_{t1}, \dots, z_{t1+r-1}}(K_{t1}, \dots, K_{t1+r-1}) = 0, \\ F_{z_{t2}, \dots, z_{t2+r-1}}(K_{t2}, \dots, K_{t2+r-1}) = 0, \\ F_{z_{t3}, \dots, z_{t3+r-1}}(K_{t3}, \dots, K_{t3+r-1}) = 0, \\ \dots \\ F_{z_{tN}, \dots, z_{tN+r-1}}(K_{tN}, \dots, K_{tN+r-1}) = 0. \end{cases} \quad (1)$$

Каждое K_t и Q_t можно выразить через K и Q_0 , в результате, получаем, систему с неизвестными, обозначающими биты в S_0 , решая которое, можно определить S_0 , то есть найти ключ ключевого потока.

Далее приведем определения и теоремы из [1], необходимые для нахождения решений системы уравнений (1); $k[x_1, \dots, x_n]$ – множество всех полиномов от переменных $x_1, \dots, x_n$ с коэффициентами над полем k с операциями сложения и умножения полиномов. Подмножество I множества $k[x_1, \dots, x_n]$ называется идеалом, если выполнены следующие условия:

1. $0 \in I$;
2. если $f, g \in I$, то $f + g \in I$;
3. если $f \in I$ и $h \in k[x_1, \dots, x_n]$, то $h \cdot f \in I$.

Идеал, порожденный функциями f и g , обозначается через $\langle f, g \rangle$.

Обозначим моном через $\chi^\alpha = \chi_1^{\alpha_1} \cdot \dots \cdot \chi_n^{\alpha_n}$, где $\alpha = (\alpha_1, \dots, \alpha_n) \in Z_{\geq 0}^n$ -векторы показателей степеней. Так как существует взаимно однозначное соответствие между мономами $\chi^\alpha = \chi_1^{\alpha_1} \cdot \dots \cdot \chi_n^{\alpha_n}$ и n -наборами показателей степеней $\alpha = (\alpha_1, \dots, \alpha_n) \in Z_{\geq 0}^n$, то упорядочение, которое мы определим на $Z_{\geq 0}^n$, определит и упорядочение на множестве мономов: если $\alpha > \beta$ в $Z_{\geq 0}^n$, то будем говорить, что $\chi^\alpha > \chi^\beta$. Определим градуированное обратное лексикографическое упорядочение на $Z_{\geq 0}^n$, пусть $\alpha, \beta \in Z_{\geq 0}^n$, тогда говорим, что $\alpha > \beta$, если

$$|\alpha| = \sum_{i=1}^n \alpha_i > |\beta| = \sum_{i=1}^n \beta_i$$

или $|\alpha| = |\beta|$ и самая правая ненулевая координата вектора $\alpha - \beta \in Z_{\geq 0}^n$ отрицательна.

Упорядочение мономов называется линейным, если для любой пары мономов χ^α и χ^β выполняется ровно одно из следующих соотношений: $\chi^\alpha > \chi^\beta$, $\chi^\alpha = \chi^\beta$, $\chi^\alpha < \chi^\beta$.

При умножении монома на полином может нарушиться порядок членов, вследствие этого могут возникнуть трудности в алгоритме деления в $k[x]$, так как произведение старшего члена полинома на моном может не быть старшим членом произведения. Поэтому требуется, чтобы упорядочение мономов обладало следующим дополнительным свойством: если $\chi^\alpha > \chi^\beta$, а χ^γ – произвольный моном, то $\chi^\alpha \chi^\gamma > \chi^\beta \chi^\gamma$. В терминах векторов показателей степеней, это означает, что если $\alpha > \beta$ в $Z_{\geq 0}^n$, то для любого $\gamma \in Z_{\geq 0}^n$ $\alpha + \gamma > \beta + \gamma$.

Мономиальным упорядочением на $k[x_1, \dots, x_n]$ называется любое бинарное отношение $>$ на $Z_{\geq 0}^n$, обладающее следующими свойствами:

- $>$ - линейное упорядочение на $Z_{\geq 0}^n$;
- если $\alpha > \beta$ и $\gamma \in Z_{\geq 0}^n$, то $\alpha + \gamma > \beta + \gamma$;
- $>$ вполне упорядочивает $Z_{\geq 0}^n$, т.е. любое непустое подмножество в $Z_{\geq 0}^n$ имеет минимальный элемент по отношению к $>$.

Пусть $f = \sum_{\alpha} a_{\alpha} \chi^{\alpha}$ – ненулевой полином в $k[x_1, \dots, x_n]$, и $>$ мономиальное упорядочение, $a_{\alpha} \in k$:

- $LT(f)$ - старший член полинома f ;

- $LT(f) = a_{\alpha_1} \chi^{\alpha_1}$, где $\alpha_1 = \max(\alpha \in Z_{\geq 0}^n: a_{\alpha} \neq 0)$,

где максимум берется по отношению к $>$;

- $LC(f) = a_{\alpha_1}$.

Пусть задано мономиальное упорядочение; конечное подмножество $G = \{g_1, \dots, g_s\}$ элементов идеала I называется базисом Гребнера идеала, если $\langle LT(g_1), \dots, LT(g_s) \rangle = \langle LT(I) \rangle$.

$\{g_1, \dots, g_s\}$ тогда и только тогда является базисом Гребнера идеала I , если старший член любого элемента из I делится на хотя бы один из $LT(g_1), \dots, LT(g_s)$.

Редуцированным базисом Гребнера полиномиального идеала называется его базис Гребнера G , такой, что

- $LC(p) = 1$ для всех $p \in G$;

- никакой моном никакого $p \in G$ не принадлежит $\langle LT(G - \{p\}) \rangle$.

Пусть дана система уравнений над полем F_q :

$$\begin{cases} f_1(x_1, \dots, x_n) = 0, \\ f_2(x_1, \dots, x_n) = 0, \\ \dots \\ f_l(x_1, \dots, x_n) = 0. \end{cases} \quad (2)$$

Теорема[5].

Пусть дан идеал, составленный из уравнений над полем F_q из системы (2):

$$I = \langle f_1, \dots, f_l, x_1^q - x_1, x_2^q - x_2, \dots, x_n^q - x_n \rangle.$$

Для любого упорядочения редуцированный базис Гребнера имеет вид:

- $\{x_1 - \chi_1, x_2 - \chi_2, \dots, x_n - \chi_n\}$, где $(\chi_1, \chi_2, \dots, \chi_n)$ – решение системы уравнений (2)

- $\{1\}$, если система (2) не имеет решений.

Таким образом, найдя редуцированный базис Гребнера системы (1), составленной из Z - функций, находим её решение, т.е. ключ ключевого потока (начальное состояние (l,m) -комбайнера). При декодировании, зная ключ ключевого потока и соотношения для формирования ключевого потока, генерируем ключевой поток, выполняем операцию XOR над шифротекстом и ключевым потоком и определяем исходное информационное сообщение.

Разработанное программное средство базируется на C++, формирует систему алгебраических уравнений. Для нахождения редуцированного базиса Гребнера в данной работе использована программа Reduce, пакет ViBas программы Reduce вычисляет редуцированный базис Гребнера идеала из многочленов в поле F_2 [4].

В таблице 1 приведены максимальные значения параметров l , m (l,m) -комбайнера, при которых возможно провести анализ поточного шифра с помощью созданной программы и программы Reduce. Анализ проводился при известных значениях первых нескольких бит ключевого генератора, соотношениях для формирования РСЛОС, функциях выхода и изменения состояния памяти (l,m) -комбайнера (в поле F_2).

Таблица 1 – Результаты исследования

Параметры l, m (l, m) – комбайнера	Результат нахождения значений элементов S_0
(2,1)	1
(2,2)	1
(2,3)	1
(2,5)	1
(2,11)	1
(2,12)	0
(3,0)	1
(3,1)	1
(3,5)	1
(3,7)	1
(3,8)	0
(4,3)	1
(4,4)	0
(5,0)	$i+p+s+1 = 0, o+a+d+1 = 0$ остальные элементы найдены точно
(5,1)	0
(7,0)	$y+i+p+s+f = 0, u+o+a+d+g = 0$. остальные элементы найдены точно
(8,0)	0

Значение «1» в столбце «Результат нахождения значений элементов S_0 » таблицы 1 означает, что найдены все элементы в S_0 , значение 0 – не найден ни один элемент в S_0 .

Таким образом, с помощью разработанной программы и программы Reduce найдено первоначальное состояние (l, m) -комбайнера (ключ для построения ключевого потока поточного шифра). В результате проведения ряда программных экспериментов по анализу ключевого потока поточного шифра установлено, что наибольшие значения

параметров l, m (l,m) -комбайнера, при которых можно найти программными способами (а именно с помощью разработанной программы на языке C++ и программы Reduce) его начальное состояние S_0 при известных значениях первых нескольких бит ключевого генератора в поле F_2 , соотношениях для формирования РСЛОС, функциях выхода и изменения состояния памяти (l,m) -комбайнера (в поле F_2) равны:

$$l=2, m=11,$$

$$l=3, m=7,$$

$$l=4, m=3,$$

при $l=5, m=0$ и при $l=7, m=0$ можно сделать вывод о некоторых элементах S_0 .

При больших параметрах, чем указанные выше, анализ поточных шифров при вышеперечисленных известных данных с помощью системы алгебраических уравнений неэффективен.

Список литературы

1. Кокс Д., Литл Дж., О'Ши Д. Идеалы, многообразия и алгоритмы. Введение в вычислительные аспекты алгебраической геометрии и коммутативной алгебры : пер. с англ. М.: Мир, 2000. 687 с.
2. Armknecht F. Algebraic attacks on certain stream ciphers: Inauguraldissertation zur Erlangung des akademischen Grades eines Doktors der Naturwissenschaften / Universitat Mannheim. Mannheim, 2006.
3. Courtois N. Algebraic attacks on combiners with memory and several outputs. Korea: ICISC, 2004.
4. Documentation Reduce. Режим доступа: <http://www.reduce-algebra.com/packages.htm> (дата обращения 20.01.2013).
5. Shannon C. Communication theory of secrecy systems // Bell Systems Techn. Journal. 1949. Vol. 28, № 4. pp. 656-719.
6. Stinson D. Cryptography. Theory and Practice. USA: CRC Press, Inc., 1995. 434 p.

Analysis of stream ciphers by solving a system of algebraic equations

03, March 2013

DOI: 10.7463/0313.0546388

Chilikov A.A., Huzina E.I.

Bauman Moscow State Technical University, 105005, Moscow, Russian Federation

chilikov@passware.comx47y47z47@mail.ru

The authors research the method of algebraic attack on stream ciphers based on solution of algebraic equations. The method is applied to a stream cipher constructed according to the disposable pad scheme. The authors consider types of key stream generators needed to construct stream ciphers. The key stream is a pseudo-random bit sequence which is combined with the original message to get the ciphertext. The algebraic method of finding the key of the key stream is described in the article. A program that generates a key stream and a system of equations to analyze the stream cipher was developed. The authors found maximum parameters of the stream cipher, the analysis of which can be carried out by software in several first elements of the key stream, its output function and changes of memory state.

Publications with keywords: [stream cipher](#), [Groebner basis](#), [linear feedback shift register](#)

Publications with words: [stream cipher](#), [Groebner basis](#), [linear feedback shift register](#)

References

1. Cox D.A., Little J., O'Shea D. *Ideals, Varieties, and Algorithms. An Introduction to Computational Algebraic Geometry and Commutative Algebra*. 2nd ed. Springer, 1997. (Ser. Undergraduate Texts in Mathematics). (Koks D., Litl Dzh., O'Shi D. *Idealy, mnogoobrazii i algoritmy. Vvedenie v vychislitel'nye aspekty algebraicheskoi geometrii i kommutativnoi algebrы*. Moscow, Mir, 2000. 687 p.)
2. Armknecht F. *Algebraic attacks on certain stream ciphers. Inauguraldissertation zur Erlangung des akademischen Grades eines Doktors der Naturwissenschaften*. Universitat Mannheim, Mannheim, 2006. (in German)
3. Courtois N. *Algebraic attacks on combiners with memory and several outputs*. Korea, ICISC, 2004.
4. *Documentation Reduce*. Available at: <http://www.reduce-algebra.com/packages.htm> , accessed 20.01.2013.
5. Shannon C. Communication theory of secrecy systems. *Bell Systems Techn. Journal*, 1949, vol. 28, no. 4, pp. 656-719.
6. Stinson D. *Cryptography. Theory and Practice*. USA, CRC Press, Inc., 1995. 434 p.